Relational Kernel-based Grasping with Numerical Features

Laura Antanas, Plinio Moreno, and Luc De Raedt

Department of Computer Science, Belgium
{laura.antanas,plinio.moreno,luc.deraedt}@cs.kuleuven.be

Abstract. Object grasping is a key task in robot manipulation. Performing a grasp largely depends on the object properties and grasp constraints. This paper proposes a new statistical relational learning approach to recognize graspable points in object point clouds. We characterize each point with numerical shape features and represent each cloud as a (hyper-) graph by considering qualitative spatial relations between neighboring points. Further, we use kernels on graphs to exploit extended contextual shape information and compute discriminative features which show improvement upon local shape features. Our work for robot grasping highlights the importance of moving towards integrating relational representations with low-level descriptors for robot vision. We evaluate our relational kernel-based approach on a realistic dataset with 8 objects.

Keywords: robot grasping, graph-based representations, numerical shape features, relational kernels, numerical feature pooling

1 Introduction

To operate in the real world, a robot requires good manipulation skills. A good robot grasp depends on the specific manipulation scenario, and essentially on the object properties, as well as grasp constraints (e.g., gripper configuration, environmental restrictions). As in robot manipulation objects are widely described using point clouds, robot grasping often relies on finding good mappings between gripper orientations and object regions (or points). To this end, much of the current work on robot grasping focuses on adapting low-level descriptors popular in the computer vision community (i.e., shape context) to characterize the graspability of an object point. Essentially, this translates into calculating, for each point in the cloud, a shape feature descriptor that summarizes a limited neighbouring surface around the point. However, such local shape features do not work properly on very complex or (self-) occluded objects.

A first contribution of this paper is to investigate whether *the structure of the object can improve robot grasping by means of statistical relational learning (SRL)*. In order to do so, we propose to employ a graph-based representation of the object that exploits both local numerical shape features and higher-level information about the structure of the object. Given a 3D point cloud of the object, we characterize each point with shape features and represent the cloud

as a (hyper-) graph by adding symbolic spatial relations that hold among neighboring object points. As a result, graph nodes corresponding to object points are characterized by distributions of numerical shape features instead of semantic labels. The derived relational graph captures extended contextual shape information of the object which may be useful to better recognize graspable points. As an example, consider a graspable point on the rim of a cup. Although it may be characterized by a misleading local shape descriptor due to its position or perceptual noise, this can be corrected by nearby graspable points with more accurate shape features.

As a second contribution, we propose a *new relational kernel-based approach to numerical feature pooling* for robot grasping. To recognize graspable points we employ relational kernels defined on the attributed graph. For each point, our relational kernel exploits extended contextual information and aggregates (or pools) numerical shape features according to the graph structure, yielding more discriminative features. Its benefit is shown experimentally on a realistic dataset. Our work highlights the importance of moving towards integrating relational representations with low-level descriptors for robot vision.

We proceed as follows. We first explain in Section 2 the grasping primitives that define our setup. Afterwards, we present our relational formulation for the learning problem considered (Section 3) and show how we solve it with variants of relational kernels (Section 4). Next, in Section 5 we present our experimental results. Before concluding, we review related work on robot grasping, feature pooling and graph kernels (Section 6).

2 The Robot Grasping Scenario and Grasping Primitives

We consider the robot scenario in Fig. 1. The robotic platform is next to a table and on the table there are one or more objects for grasping exploration. The robot has the following components: a mobile component, an arm, a gripper and a range camera. An object (e.g., cup, glass) may be placed on the table at various poses. Each pose provides a point cloud, obtained via the range sensor. The points above the table are converted, using segmentation techniques (e.g., [24]), into a point cloud describing the object. Fig. 1 illustrates the point cloud of the visible side of a cup placed on the table sideways. The goal is to determine the pre-grasp pose, that is, where to place the gripper with respect to the object in order to execute a stable grasp. Motion planning from the current gripper pose to the pre-grasp pose reduces the number of grasping hypotheses due to kinematic and environmental constraints. The reduced set of reachable local regions provides the data samples for learning to recognize graspable object points.

We consider three types of domain primitives which we use to build our relational representation (or hyper-graphs) of the grasping problem: *reaching points*, their *3D locations* and their numerical *shape features*. Reaching points are labeled using the simulator. The robot executes grasps on the object points and if they are successful, the reaching points become positive instances. Next,

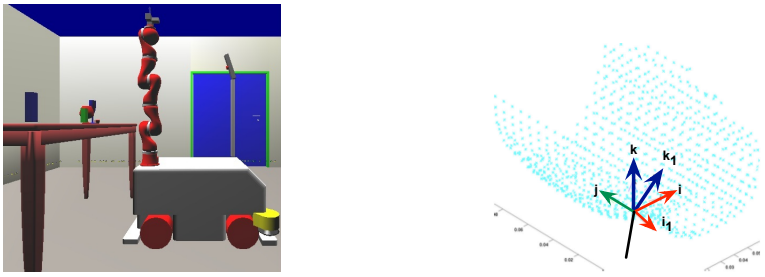


Fig. 1. Robot grasping scenario. The gripper and objects on the table (left). A partial point cloud of a can placed on the table (right). The (i, j, k) is the reference frame of the camera centred at the sample point. Its normal is the black line. The (i_1, j, k_1) is the reference frame of the 3D grid, which is obtained by rotating the (i, j, k) frame along the y axis.

each reaching point is characterized by several local 3D shape features computed in its neighborhood. The neighborhood of each point consists of a 3D grid centred at the reaching point and oriented with respect to the projection of the points normal on the table plane and the gravity vector, as illustrated in Fig. 1. We consider as neighborhood grid, in turn, a gripper cell and a sphere around the point and calculate three shape features: 3D shape context (SC) [18], point feature histogram (PFH) [27] and viewpoint feature histogram (VFH) [28].

While the PFH feature encodes the statistics of the shape of a point cloud by accumulating the geometric relations between all point pairs, the VFH augments PFH with the relation between the camera's point of view and the point cloud of an object. The 3D SC describes the shape of the object as quantitative descriptions centered at different points on the surface of the object. The shape context of a point is a coarse histogram of the relative coordinates of the remaining surface points. The bins of the histogram are constructed by the overlay of concentric shells around the center point and sectors emerging from this point.

3 Relational Grasping: Problem Formulation

Next, we represent the grasping primitives as a relational database and use it as input to our relational learning system. We use the kLog framework [11] to build our relational kernel-based approach to grasping point recognition. Embedded in Prolog, kLog is a domain specific language for kernel-based learning, that allows to specify in a declarative way relational learning problems. It learns from interpretations [8], transforms the relational databases into graph-based representations and uses graph kernels to extract the feature space.

Fig. 2 illustrates the information flow in kLog for robot grasping. We model our graspable point recognition problem starting from the grasping primitives which we represent as relational databases. Next, we define declaratively spatial relations between reaching points. The extended relational database is used by

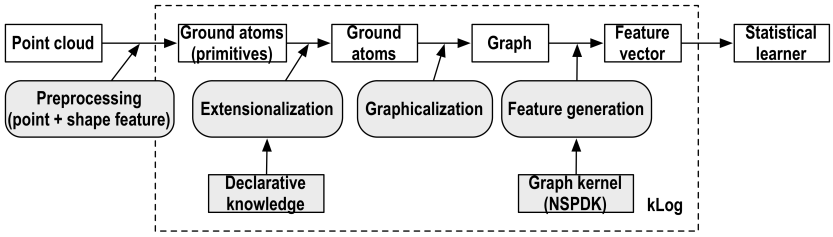


Fig. 2. From point clouds to feature vectors in kLog.

kLog to build kernel features which are finally used for learning. We explain in more detail each step for our grasping problem.

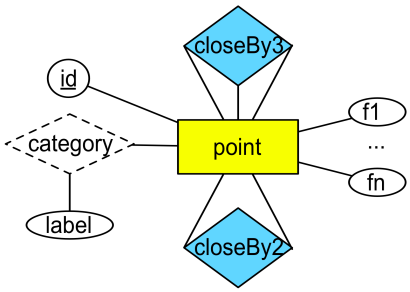
3.1 Data modeling

Grasping primitives are represented at a higher level using a relational language derived from its associated entity/relationship (E/R) data model, as in database theory [12], with some further assumptions required by kLog. It is based on entities, relationships linking entities and attributes that describe entities and relationships. Fig. 3(a) shows the E/R diagram for our grasping point problem. A *reaching entity* is any reaching point. It is represented by the relation $\texttt{point}(id, f_1, \ldots, f_n)$, which indicates that it has a unique identifier id (underlined oval) and shape properties. The vector $[f_1, \ldots, f_n]$ represents a shape feature characterizing the reaching point. Each f_i is a shape feature vector component and is represented as an entity attribute. For example, the tuple $\texttt{point}(p_1, 10.8, \ldots, 557.9)$ specifies a specific reaching point entity (depicted as rectangle in Fig. 3(b)), where p_1 is its identifier and the other arguments are shape feature components.

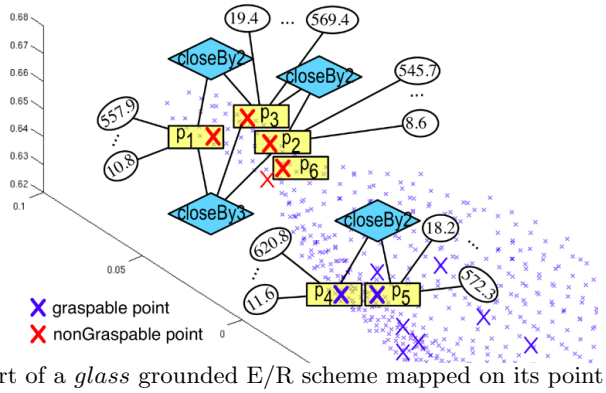
Relationships are qualitative *spatial relations* among entities (diamonds) and are derived from their 3D spatial locations. They impose a structure on reaching entities. In practice, we employ the relationship $\texttt{closeBy2}(p_1, p_3)$ which indicates that reaching entities p_1 and p_3 are spatially close to each other, and the relationship $\texttt{closeBy3}(p_1, p_2, p_3)$ which indicates that reaching entities p_1, p_2 and p_3 are spatially close to each other. A special relationship is introduced by the predicate $\texttt{category}(id, class)$ (dashed diamond). It is linked to reaching entities and associates a binary class label $grasp/nonGrasp$ to each entity, indicating if the reaching point is graspable or not.

3.2 Declarative and Relational Feature Construction

We define the spatial relations using logical rules. For example, the relation $\texttt{closeBy2}/2$ holds between two points that belong to the same point cloud and are spatially close to each other. It can be defined as follows:



(a) Proposed E/R scheme: rectangles denote entity vertices, diamonds denote relationships, and circles (except point id) denote local properties.



(b) Part of a *glass* grounded E/R scheme mapped on its point cloud.

$x = \{\texttt{point}(p_1, 10.8, \ldots, 557.9), \texttt{point}(p_2, 8.6, \ldots, 545.7), \texttt{point}(p_3, 19.4, \ldots, 569.4),$
$\texttt{point}(p_4, 11.6, \ldots, 620.8), \texttt{point}(p_5, 18.2, \ldots, 572.3), \ldots, \texttt{closeBy2}(p_1, p_3),$
$\texttt{closeBy2}(p_3, p_2), \texttt{closeBy2}(p_4, p_5), \ldots, \texttt{closeBy3}(p_1, p_2, p_3), \ldots\}.$
$y = \{\texttt{category}(p_1, nonGrasp), \texttt{category}(p_2, nonGrasp), \texttt{category}(p_3, nonGrasp),$
$\texttt{category}(p_4, grasp), \texttt{category}(p_5, grasp), \ldots\}.$

(c) Point cloud interpretation $i = (x, y)$ of a *glass* point cloud.

Fig. 3. Relational robot grasping in kLog.

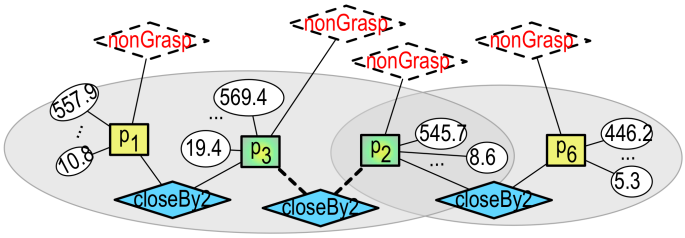


Fig. 4. From point cloud graph to feature vectors in kLog.

$$\begin{aligned}\texttt{closeBy2}(P_1, P_2) \leftarrow\ & \texttt{point}(P_1, F_{11}, \ldots, F_{1n}), \texttt{point}(P_2, F_{21}, \ldots, F_{2n}),\\ & \texttt{cloud}(P_1, V), \texttt{cloud}(P_2, V), P_1 \leq P_2,\\ & \texttt{objectLength}(L), \texttt{objectHeight}(H), \texttt{objectWidth}(W),\\ & T_x = c * L, T_y = c * H, T_z = c * W,\\ & \texttt{edist}(P_1, P_2, D_x, D_y, D_z), D_x < T_x, D_y < T_y, D_z < T_z.\end{aligned}$$

The condition $\texttt{cloud}(P_1, V)$, $\texttt{cloud}(P_2, V)$ specifies that P_1 and P_2 belong to the same point cloud V. The inequality $P_1 \leq P_2$ removes the symmetry of the close by relation. The relation $\texttt{edist}/5$, defined in a similar way, represents the normalized Euclidian distance between 2 points in the 3D space. As the definition shows, it is projected on all 3 axes and thresholded on each axis i. The thresholds T_i are distance thresholds calculated for every object from the object dimensions using a constant ratio c.

The close by relation defined above allows cycles of size 3 or greater in the graph. We enforce more sparsity by allowing the $\texttt{closeBy2}/2$ relation between 2 points to hold if there does not exist another path between the two points that involves another node, thus, allowing only cycles of minimum size 4. We use the sparser close by relation in practice as it gives better results. If we denote the previous $\texttt{closeBy2}/2$ relation as $\texttt{closeBy2\_initial}/2$, the sparser relation is defined as:

$$\texttt{closeBy2}(P_1, P_2) \leftarrow \texttt{closeBy2\_initial}(P_1, P_2), L = 3, \texttt{no path}(P_1, P_2, L).$$

The relation $\texttt{path}/3$ checks if there is a path smaller than or equal to 2 edges between nodes P_1 and P_2.

We define in a similar way the relation $\texttt{closeBy3}/3$ which holds between three points that belong to the same point cloud and are spatially close to each other.

In our setting each point cloud is represented as an instance of a relational database (i.e., as a set of relations), and thus, as a *point cloud interpretation*. Object point clouds are assumed to be independent. An example of a point cloud interpretation is given in Fig. 3(c).

3.3 The Relational Problem Definition

We formulate the learning problem at the relational representation level in the following way: given a training set $D = \{(x_1, y_1), \ldots, (x_2, y_2), \ldots, (x_m, y_m)\}$ of m independent interpretations, the goal is to learn a mapping $h : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X}$ denotes the set of all points x_i^k in any point cloud interpretation i, with $i \in \{1, \ldots, m\}$ and $\mathcal{Y}$ is the set of target atoms y_i^k. The pair $e^k = (x_i^k, y_i^k)$ is a training example, where $k \in \{1, \ldots, n\}$ and n is the number of training instances in the point cloud interpretation i. One training example e^k is, thus, a smaller interpretation, part of the larger point cloud interpretation, and corresponds to one point in the object point cloud. Given a new point in a point cloud interpretation we can use h to predict its target category $\texttt{category}/2$.

3.4 Graphicalization

Next, each interpretation x is converted into a bipartite graph G which introduces a vertex for each ground atom. Vertices correspond to either entities or relationships, but identifiers are removed. Edges connect entities with relationships. Fig. 3(b) shows part of the graph mapped on a point cloud. The graph is the result of grounding the E/R diagram for a particular point cloud.

4 Relational Kernel Features

We solve the grasping recognition problem in a supervised learning setting. We employ two variants of the fast neighborhood subgraph pairwise distance kernel [7]. The kernel is a decomposition kernel [13] that counts the number of common "parts" between two graphs. In our case the graph represents the contextual shape information of one point in the point cloud. The decomposition kernel between two graphs is defined with the help of relations $R_{r,d}$ ($r = 0, \ldots, R$ and $d = 0, \ldots, D$) as follows:

$$K(G, G') = \sum_{r=0}^{R} \sum_{d=0}^{D} \sum_{\substack{A,\, B \,\in\, R_{r,d}^{-1}(A,\, B,\, G) \\ A',\, B' \,\in\, R_{r,d}^{-1}(A',\, B',\, G')}} \kappa((A, B), (A', B')) \tag{1}$$

where $R_{r,d}^{-1}(A, B, G)$ returns the set of all pairs of neighborhoods (or balls) (A, B) of radius r with roots at distance d that exist in G. Thus, a "part" is a pair of neighborhoods (or a pair of balls). Fig. 4 shows a neighborhood-pair feature with $R = 2$ and $D = 2$ for our grasping problem. The kernel hyper-parameters maximum radius R and maximum distance D are set experimentally. We ensure that only neighborhoods centered on the same type of vertex will be compared, constraint imposed by the equation:

$$\kappa((A, B), (A', B')) = \kappa_{root}((A, B), (A', B')) \cdot \kappa_{subgraph}((A, B), (A', B')), \tag{2}$$

where the component $\kappa_{root}((A, B), (A', B'))$ is 1 if the neighborhoods to be compared have the same type of roots, while the component $\kappa_{subgraph}((A, B), (A', B'))$ compares the pairs of neighborhood graphs extracted from two graphs G and G'. We solve the grasping problem using two specializations of $\kappa_{subgraph}$. Because we deal both with symbolic and numerical attributed graphs, we employ a hard-soft variant which combines an exact matching kernel for the symbolic relations and a soft match kernel for numerical properties of the relations, and a soft variant which uses only a soft match kernel.

Soft matching The soft matching kernel uses the idea of multinomial distribution (i.e., histogram) of labels. It discards the structural information inside the graph. Contextual information is still incorporated by the (sum) pooling operation applied on the numerical properties of the points.

$$\kappa_{subgraph}((A,B),(A',B')) = \sum_{\substack{v \in V(A) \cup V(B) \\ v' \in V(A') \cup V(B')}} \mathbf{1}_{\ell(v)=\ell(v')} \kappa_{tuple}(v,v') \tag{3}$$

where $V(A)$ is the set of vertices of A and $\ell(v)$ is the label of vertex v. If the atom $point(p_1, f_1, \ldots, f_c, \ldots, f_m)$ is mapped into vertex v, $\ell(v)$ returns the signature name *point*. In this case κ is decomposed in a part that counts the vertices that share the same labels $\ell(v)$ in the neighborhood pair and ensures matches between tuples with the same signature name ($\mathbf{1}_{\ell(v)=\ell(v')}$), and a second part that takes into account the tuple of property values. These are real values and thus, the kernel on the tuple considers each element of the tuple independently with the standard product:

$$\kappa_{tuple}(v,v') = \sum_c prop_c(v) \cdot prop_c(v') \tag{4}$$

where for the atom $\texttt{point}(p_1, f_1, \ldots, f_c, \ldots, f_m)$, mapped into vertex v, $prop_c(v)$ returns the property value f_c. In words, the kernel will count the number of symbolic labels and will sum property values that belong to vertices with same labels $l(v)$ that are contained in the neighborhood pair.

Hard-soft matching The hard-soft variant replaces the label $l(v)$ in Equation 3 with a relabeling procedure for the discrete signature names. We proceed with a canonical encoding that guarantees that each vertex receives a label that identifies it in the neighborhood graph based on the exact extracted structure of the ball with respect to the relabeled vertex. Then, the exact match kernel for the discrete part is defined as $\kappa_{subgraph}((A,B),(A',B')) = 1$ iff (A,B) and (A',B') are pairs of isomorphic graphs. The isomorphism is ensured by the vertices canonical relabeling. This match ensures that the contextual structure of the subgraphs matched is the same. Concerning the real valued properties, we use the standard product as in Equation 4 for the tuples of vertices with the same relabelings. The spatial relations injected in the graph and its structure ensure that the pooled features are the ones belonging to vertices with a similar relabeling. In this way, we only sum the features with same contextual structure.

There are several advantages of using kLog and its kernel-based language. First, it can take relational contextual features into account in a principled way. Second, it allows fast computations with respect to the interpretation size, which allows us to explore different measures of contextual information via the kernel hyper-parameters. Third, it provides a flexible architecture in which only the specification language for relational learning problems is fixed. Actual features are determined by the choice of the graph kernel. In this setting, experimenting with alternative feature spaces is rapid and intuitive. For more details, see [11].

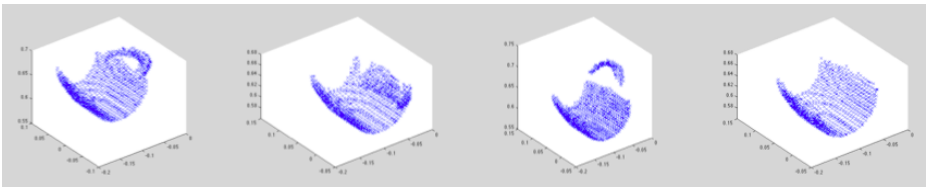


Fig. 5. Point clouds representing partial views of a cup.

5 Experiments

We evaluate whether our relational kernel-based approach can exploit contextual shape information by pooling numerical features. Specifically, we investigate the following questions:
(Q1) Does numerical shape feature pooling improve upon local shape features for the robot grasping task considered?
(Q2) Does hard-soft matching improve over soft matching when incorporating contextual shape information?

To answer these questions, we perform experiments with all shape features considered in turn.

5.1 Dataset and Evaluation

We consider a realistic dataset similar to that in [23]. It is gathered using 8 objects: ellipsoidal object, rectangular object, round object, 2 glasses and 3 cups. The dataset contains 2631 instances (1972 positives and 659 negatives) and it was obtained in the ORCA simulation environment [3]. To gather the dataset, the robot performed grasping trials on a large number of reaching points. The setup is shown in Fig. 1 (left).

The goal is to evaluate the performance of our approach across the different objects considered. We estimate it under partial views, that is, each object is characterized by several partial point clouds, one for each view. The number of views can differ from object to object. Fig. 5 shows four views for one of the cups. In practice, all views belonging to the same object are mapped to one interpretation, and thus, one interpretation corresponds to one object. Because the views are not spatially aligned, we consider spatial relations only between points that belong to the same view.

For performance evaluation, we apply the leave-one-out CV method where one object is used for testing and the rest for training. In all our experiments we used a SVM with a linear kernel on top of the relational kernel features. The SVM cost parameter was set to 1. Because the dataset is unbalanced (with more positives than negatives), we evaluate performance in terms of the area under the ROC curve (AUC) and the area under the precision-recall curve (AP) which are not sensitive to the distribution of instances to classes. We also report the

true positive rate (TPR) and accuracy (Acc) for both datasets. In order to better cope with the unbalanced data, the SVM implementation used (LIBSVM [6]) assigns different weights to positive and negative instances. In our case, we assign more weight to the negatives. The weight is selected using the leave-one-out CV for each feature type (when no relations are used), and is kept the same for that feature as we gradually add relations.

5.2 Results and discussion

In the following we present quantitative experimental results for both questions[1]. Results in bold font indicate the best performance. For each feature type we start with local feature vectors and we gradually add the different relations considered, `closeBy2`/2 and `closeBy3`/3, respectively. As a baseline for comparison we use the local feature vectors alone, without any spatial relations. We also present results with all available features (VFH + PFH + SC = *all*) in one experiment with and without relations. We report performance results using the hard-soft matching kernel in Table 1 for sphere features and cell features setups. They are obtained for hyper-parameters R=2, D=6 (for *shape feature+closeBy2* settings) or R=2, D=8 (for the rest of the settings). The results in italics mark the best results for each local feature type (i.e.,VFH/PFH/SC/all) in each grasping settings (gripper cell/sphere). The results in bold mark the best results for each grasping setting across all feature types considered. They show that the use of qualitative relations to pool features improves robot grasping performance for all shape feature types considered. This answers positively (Q1).

We answer question (Q2) by plotting the ROC curves for both soft and hard-soft kernels for sphere and cell features. The results in Fig. 6(a) and (b) show that hard-soft matching improves considerably upon soft matching. The curves correspond to hyper-parameters $R = 2$, $D = 8$ and `closeBy2`/2 + `closeBy`/3 relation, which give the best performance. Thus, contextual structure in the point cloud is highly relevant and ensures pooling the right numerical shape features.

6 Related work

In visual recognition a number of feature extraction techniques based on image descriptors (e.g., SIFT) have been proposed. They usually encode the descriptors over a learned codebook and then summarize the distribution of the codes by a pooling step [5, 14]. While the coding step produces representations that can be aggregated without losing too much information, pooling these codes gives robustness only to small transformations of the image. One fact that makes the coding step necessary in standard computer vision tasks is that image descriptors such as SIFT cannot be pooled directly with their neighbours without losing information. Differently, our contribution for robot grasping considers shape feature pooling without the coding step, by means of SRL techniques.

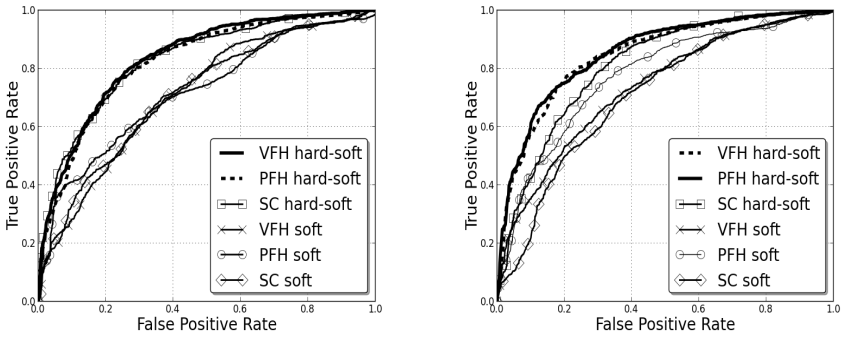
[1] These results contain an errata to the results reported in [20].

	AUC		AP		Acc (%)		TPR (%)	
Shape features	sphere	cell	sphere	cell	sphere	cell	sphere	cell
VFH	0.66	0.70	0.83	0.86	60.43	75.33	58.52	*89.50*
VFH+closeBy2	0.80	0.83	0.92	0.93	72.33	80.16	73.33	89.45
VFH+closeBy3	0.83	0.84	***0.93***	*0.94*	78.91	81.45	82.05	88.74
VFH+closeBy2+closeBy3	***0.84***	*0.85*	***0.93***	*0.94*	*79.32*	*81.91*	*82.51*	89.35
PFH	0.70	0.71	0.88	0.86	60.62	76.24	56.80	91.13
PFH+closeBy2	0.80	0.83	0.92	0.93	73.20	79.82	74.29	90.11
PFH+closeBy3	0.82	0.85	***0.93***	*0.94*	77.77	82.25	80.68	91.08
PFH+closeBy2+closeBy3	*0.83*	***0.86***	***0.93***	*0.94*	*77.99*	***83.01***	*81.03*	*91.63*
SC	0.75	0.72	0.88	0.85	73.58	69.14	78.85	71.35
SC+closeBy2	0.80	0.80	0.92	0.90	76.09	77.27	81.29	82.86
SC+closeBy3	*0.83*	*0.81*	***0.93***	*0.91*	79.29	78.94	84.08	83.92
SC+closeBy2+closeBy3	*0.83*	*0.81*	***0.93***	*0.91*	***79.55***	*79.82*	***84.13***	*85.14*
all	0.75	0.71	0.89	0.86	74.15	76.66	80.02	92.55
all+closeBy2	0.81	0.84	***0.93***	0.94	74.15	81.53	78.35	***92.60***
all+closeBy3	*0.83*	***0.86***	***0.93***	***0.95***	78.72	82.25	83.06	91.99
all+closeBy2+closeBy3	*0.83*	***0.86***	***0.93***	***0.95***	*79.29*	*82.82*	*83.92*	92.29

Table 1. Hard-soft matching results for sphere and gripper cell setups.

Previous works on visual-dependent robot grasping have shown promising results on learning grasping points from image-based 2D descriptors [21,29]. Other works exploit combinations of image-based and point cloud-based features [4,15]. Saxena et. al. [29] propose to infer grasping probabilities from image filter responses at the object points. Their approach allows to discriminate graspable from non-graspable points and transfer knowledge to new objects. However, it does not consider the parameters of the gripper to estimate the quality of the grasping. Jiang et. al. [15] extend this approach by computing grasping stability features from the point clouds. In their method, the point cloud features are linked to the gripper configuration, while the image-based features are linked to the visual graspability of a point. Differently, we consider dense 3D data for both gripper configuration and visual graspability. Kraft et. al. [16,17] propose to learn by exploration graspable points of an object. Nevertheless, their learning procedure is specific to each object, and it is difficult to transfer the skills learned to other objects. A major difference is that we learn with features that generalize across objects.

Furthermore, a significant number of vision-based grasping methods learn mappings from 2D/3D features to grasping parameters [4, 19, 22, 30]. However, it turns out that it is difficult to link a 3D gripper orientation to local shape features without considering contextual or global object information. Only recently, methods that take global and symbolic information into account have been proposed [1]. They benefit from increased geometric robustness, which gives advantages with respect to the pre-shape of the robotic hand and general shape of the object, generating more accurate grasps. Nevertheless, this work relies on complete object point clouds, and object reconstruction based on



(a) sphere features (b) gripper cell features

Fig. 6. ROC curves for soft and hard-soft matching kernels; R=2,D=8; VFH/PFH/SC + closeBy2 + closeBy3.

single views is a difficult problem due to lack of observability of the self-occluded parts. Differently, our contribution to robot grasping exploits contextual shape information of objects from partial views and, additionally, we employ a new relational approach to vision-based grasping that considers symbolic and numerical attributed graphs.

From the SRL perspective, purely relational learning techniques have been previously used to learn from point clouds. The work in [9, 10] uses first-order clause inducing systems to learn from discrete primitives (e.g., planes, cylinders) classifiers for concepts such as boxes, walls, cups or stairs. Differently, we propose a SRL approach to recognize graspable points that is based on relational kernels. A related graph kernel designed for classification and retrieval of labeled graphs was employed in [25, 26]. There, in the context of robot grasping, the authors consider the tasks of object categorization and similar object view retrieval. Object graphs are obtained as k-nearest neighbor graphs from object point clouds and graph nodes are characterized by semantic labels. The kernel is an evolving propagation kernel based on continuous distributions as graph features, which are built from semantic node labels, and on a locality sensitive hashing function to ensure meaningful features. In contrast, our work focuses on recognizing graspable points in the cloud by employing a flexible decomposition kernel. It takes as input numerical shape vectors organized in graph structures, and computes graph features by pooling meaningful shape features that are ensured by the structure of the graph. In this case, we construct object graphs declaratively using relational background knowledge and we characterize graph nodes by numerical shape features instead of semantic labels. A similar graph kernel was employed in [2] for visual recognition of houses. However, there, the visual input features were discrete and did not have a continuous numerical form.

7 Conclusions

This paper proposes a relational kernel-based approach to recognize graspable object points. We represent each object as an attributed graph, where nodes corresponding to object points are characterized by distributions of numerical shape features. Extended contextual object shape information is encoded via qualitative spatial relations among object points. Next, we use kernels on graphs to compute highly discriminative features based on contextual information. We show experimentally that pooling spatially related numerical shape feature improves robot grasping results upon purely local shape-based approaches.

We point out three directions for future work. A first direction is to investigate how similar SRL techniques working directly with numerical features can help other robot vision tasks. A second direction is to validate our results on datasets that contain a wider range of object categories. Finally, a third direction is to investigate other spatial relations or domain knowledge that could give even better results for the robot grasping problem considered.

References

1. Aleotti, J., Caselli, S.: Part-based robot grasp planning from human demonstration. In: ICRA. pp. 4554–4560 (2011)
2. Antanas, L., Frasconi, P., Costa, F., Tuytelaars, T., De Raedt, L.: A relational kernel-based framework for hierarchical image understanding. In: Gimel'farb, G.L., Hancock, E.R., Imiya, A.I., Kuijper, A., Kudo, M., Shinichiro Omachi, S., Windeatt, T., Yamada, K. (eds.) Lecture Notes in Computer Science, International Workshops on Structural and Syntactic Pattern Recognition and Statistical Techniques in Pattern Recognition. pp. 171–180. Springer (Nov 2012)
3. Baltzakis, H.: Orca simulator. `http://www.ics.forth.gr/cvrl/_software/orca_setup.exe` (2005)
4. Bohg, J., Kragic, D.: Learning grasping points with shape context. RAS 58(4), 362–377 (2010)
5. Boureau, Y.L., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: CVPR. pp. 2559–2566 (2010)
6. Chang, C., Lin, C.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3), 27:1–27 (2011)
7. Costa, F., De Grave, K.: Fast neighborhood subgraph pairwise distance kernel. In: ICML. pp. 255–262 (2010)
8. De Raedt, L.: Logical and Relational Learning. Cognitive Technologies, Springer (2008)
9. Farid, R., Sammut, C.: Plane-based object categorisation using relational learning. ML 94(1), 3–23 (Jan 2014)
10. Farid, R., Sammut, C.: Region-based object categorisation using relational learning. In: PRICAI 2014: Trends in Artificial Intelligence, Lecture Notes in Computer Science, vol. 8862, pp. 357–369. Springer International Publishing (2014)
11. Frasconi, P., Costa, F., De Raedt, L., De Grave, K.: klog: A language for logical and relational learning with kernels. Artificial Intelligence 217(0), 117 – 143 (2014)
12. Garcia-Molina, H., Ullman, J.D., Widom, J.: Database Systems: The Complete Book. Prentice Hall Press, Upper Saddle River, NJ, USA, 2 edn. (2008)

13. Haussler, D.: Convolution kernels on discrete structures. Tech. Rep. UCSC-CRL-99-10, University of California at Santa Cruz (1999)
14. Jia, Y., Huang, C., Darrell, T.: Beyond spatial pyramids: Receptive field learning for pooled image features. In: CVPR. pp. 3370–3377 (2012)
15. Jiang, Y., Moseson, S., Saxena, A.: Efficient grasping from rgbd images: Learning using a new rectangle representation. In: ICRA. pp. 3304–3311 (2011)
16. Kraft, D., Detry, R., Pugeault, N., Baseski, E., Guerin, F., Piater, J.H., Krüger, N.: Development of object and grasping knowledge by robot exploration. IEEE T. Autonomous Mental Development 2(4), 368–383 (2010)
17. Kraft, D., Detry, R., Pugeault, N., Baseski, E., Piater, J.H., Krüger, N.: Learning objects and grasp affordances through autonomous exploration. In: ICVS. pp. 235–244 (2009)
18. Krtgen, M., Novotni, M., Klein, R.: 3D shape matching with 3D shape contexts. In: The 7th Central European Seminar on Computer Graphics (2003)
19. Lenz, I., Lee, H., Saxena, A.: Deep learning for detecting robotic grasps. CoRR abs/1301.3592 (2013)
20. Mocanu-Antanas, L.: Relational Visual Recognition. Ph.D. thesis, Informatics Section, Department of Computer Science, Faculty of Engineering Science (2014)
21. Montesano, L., Lopes, M.: Learning grasping affordances from local visual descriptors. In: ICDL. pp. 1–6. IEEE Computer Society (2009)
22. Montesano, L., Lopes, M.: Active learning of visual descriptors for grasping using non-parametric smoothed beta distributions. Humanoids 60(3), 452–462 (2012)
23. Moreno, P., Hornstein, J., Santos-Victor, J.: Learning to grasp from point clouds. Tech. Rep. Vislab-TR001/2011, Department of Electrical and Computers Engineering, Instituto Superior Técnico, Portugal (September 2011)
24. Muja, M., Ciocarlie, M.: Table top segmentation package. `http://www.ros.org/wiki/tabletop_object_detector` (2012)
25. Neumann, M., Garnett, R., Moreno, P., Patricia, N., Kersting, K.: Propagation kernels for partially labeled graphs. In: MLG–2012 (2012)
26. Neumann, M., Moreno, P., Antanas, L., Garnett, R., Kersting, K.: Graph kernels for object category prediction in task–dependent robot grasping. In: MLG–2013 (2013)
27. Rusu, R.B.: Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. Ph.D. thesis, Computer Science Department, Technische Universitat Munchen, Germany (October 2009)
28. Rusu, R.B., Bradski, G., Thibaux, R., Hsu, J.: Fast 3D recognition and pose using the viewpoint feature histogram. In: IROS. Taipei, Taiwan (October 2010)
29. Saxena, A., Driemeyer, J., Ng, A.Y.: Robotic grasping of novel objects using vision. IJRR 27(2), 157–173 (Feb 2008)
30. Saxena, A., Wong, L.L.S., Ng, A.Y.: Learning grasp strategies with partial shape information. In: AAAI. pp. 1491–1494. AAAI Press (2008)